Shor's Algorithm and Elliptic Curve Cryptography

Ivan Murashko

Contents

Introduction

We will start with the classical construction and then look at the place where the quantum algorithm changes the situation. Elliptic curve cryptography uses a group of points on a curve. The public operation in this group is scalar multiplication. The private problem is inverse to it: given two points g and q, find a number x such that

$$xg = q.$$

This problem is called the discrete logarithm problem on elliptic curves.

Classically, the problem is considered hard for well-chosen curves and subgroups. Shor's algorithm [4] changes the character of the problem. It replaces the direct search for x with a period finding problem, and the period can be found by a quantum Fourier transform. We will use one small curve throughout the article. The numbers are not cryptographically large, but they are convenient because every step can be written explicitly.

1 Elliptic Curves over the Real Numbers

In elliptic cryptography we consider sets of objects that form a group. As such a set we will use points that belong to a curve. We start with the curve over the real numbers:

$$E : y^2 = x^3 + ax + b.$$

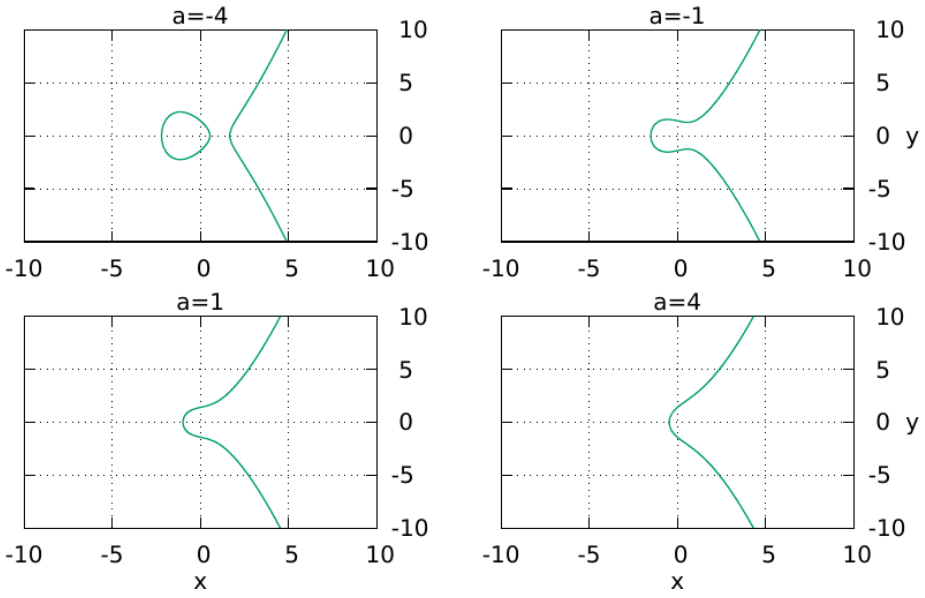

Figure 1: The elliptic curve $y^2 = x^3 + ax + 2$ over $\mathbb{R}$ for several values of a.

The coefficients a and b have to satisfy the condition

$$4a^3 + 27b^2 \neq 0.$$

This condition excludes singular curves, i.e. curves where the cubic polynomial has a multiple root [5].

We have to define a binary operation on the points of the curve. This operation maps two points p and q to a third point r:

$$p + q = r.$$

We will call this operation addition.

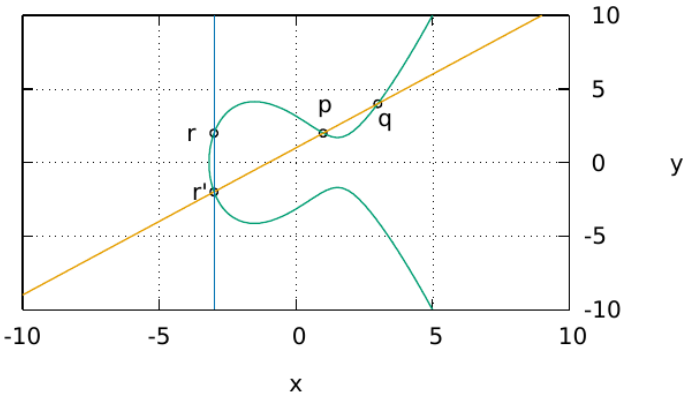

Figure 2: Addition of two points on $y^2 = x^3 - 7x + 10$ over $\mathbb{R}$. The line through $p = (1, 2)$ and $q = (3, 4)$ intersects the curve in $r' = (-3, -2)$. After reflection we obtain $r = (-3, 2)$, i.e. $p + q = r$.

Let us describe the construction. Suppose that the points p and q have coordinates (x_p, y_p) and (x_q, y_q). If $x_p \neq x_q$, then the line through these two points has slope

$$m = \frac{y_p - y_q}{x_p - x_q}.$$

This line intersects the cubic curve in one more point r'. If the coordinates of this point are $(x_{r'}, y_{r'})$, then

$$y_{r'} = y_p + m(x_{r'} - x_p).$$

The point r' belongs to the curve, therefore

$$(y_p + m(x_{r'} - x_p))^2 = x_{r'}^3 + ax_{r'} + b.$$

After moving all terms to one side we obtain a cubic equation in $x_{r'}$. Its three roots are x_p, x_q, and $x_{r'}$. Hence the coefficient of x^2 gives us

$$x_{r'} + x_p + x_q = m^2.$$

Thus

$$x_{r'} = m^2 - x_p - x_q.$$

The sum $r = p+q$ is defined as the reflection of r' with respect to the X-axis. Therefore the final formula is

$$\begin{aligned} x_r &= m^2 - x_p - x_q, \\ y_r &= -y_p + m(x_p - x_r). \end{aligned} \tag{1}$$

There are two special cases. First, if $p = q$, then the line through the two points becomes the tangent line. In this case

$$m = \frac{3x_p^2 + a}{2y_p}.$$

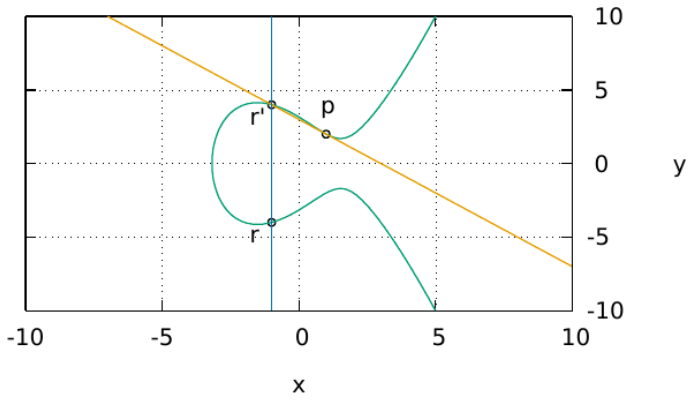

Figure 3: Point doubling. When two points coincide, the secant line is replaced by the tangent line at p.

Second, if $q = -p$, then the line is vertical. It does not produce a finite third point. For this reason we add one extra point 0, the point at infinity, and define

$$p + (-p) = 0.$$

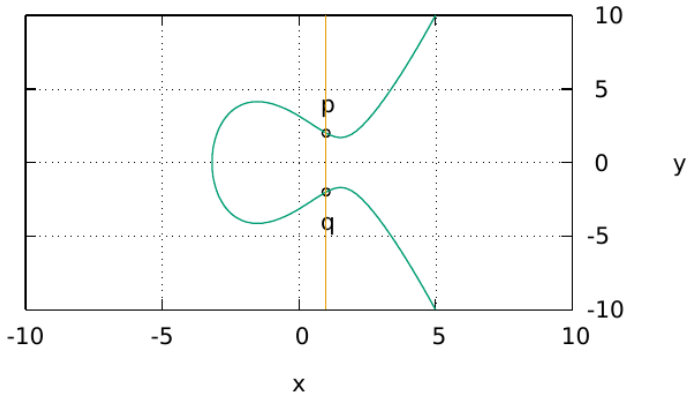

Figure 4: A vertical line through inverse points. This case gives the point at infinity: $p + (-p) = 0$.

Thus the elliptic curve over the real numbers is the set

$$E(\mathbb{R}) = \{(x, y) \in \mathbb{R} \times \mathbb{R} : y^2 = x^3 + ax + b\} \cup \{0\}.$$

With the operation described above, this set forms an abelian group. The associativity of this operation is not obvious from the geometric definition, but it is a standard property of elliptic curves [5].

Remark. At first glance this addition looks artificial. It would be more natural to add points as vectors in the plane. But vector addition does not preserve the curve: if $p, q \in E(\mathbb{R})$, it is usually not true that the vector sum also belongs to $E(\mathbb{R})$. The operation above is useful because it keeps us inside the same set and therefore gives a group.

2 Elliptic Curves over a Finite Field

The same short Weierstrass construction can be moved from $\mathbb{R}$ to fields whose characteristic is neither 2 nor 3. For cryptography, the important case in this article is the finite field $\mathbb{F}_p$, where $p > 3$ is prime. We define

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p : y^2 \equiv x^3 + ax + b \pmod{p}\} \cup \{0\}.$$

The number of points on the curve is called the order of the curve and is denoted by $|E|$.

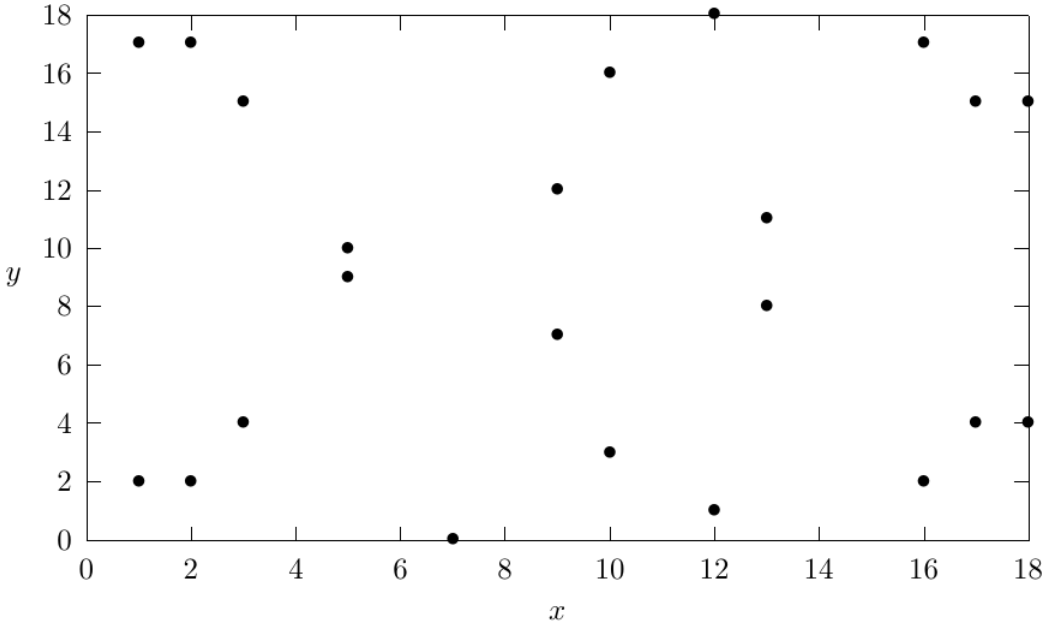

Figure 5: The curve $y^2 \equiv x^3 - 7x + 10 \pmod{19}$ over $\mathbb{F}_{19}$.

For every point $a = (x_a, y_a)$ there is an inverse point

$$-a = (x_a, -y_a \bmod p).$$

The addition law has the same form as before, but all operations are taken modulo p. If $a + b = c$ and $b \neq -a$, then

$$\begin{aligned} x_c &\equiv m^2 - x_a - x_b \pmod{p}, \\ y_c &\equiv -y_a + m(x_a - x_c) \pmod{p}. \end{aligned} \tag{2}$$

The slope is computed as follows:

$$m \equiv \begin{cases} (y_a - y_b)(x_a - x_b)^{-1} \pmod{p}, & x_a \neq x_b, \\ (3x_a^2 + a)(2y_a)^{-1} \pmod{p}, & x_a = x_b. \end{cases}$$

Here z^{-1} denotes the multiplicative inverse of z modulo p. If $b = -a$, then $a + b = 0$.

3 Scalar Multiplication and Discrete Logarithm

Let n be a natural number and let $a \in E(\mathbb{F}_p)$. We define scalar multiplication as repeated addition:

$$n \cdot a = \underbrace{a + a + \cdots + a}_{n \text{ times}}.$$

A direct implementation needs $O(n)$ additions. Using the usual divide-and-conquer idea, or double-and-add, the same operation can be computed with $O(\log n)$ additions.

Example. Consider the curve

$$E(\mathbb{F}_{19}) = \{(x, y) \in \mathbb{F}_{19} \times \mathbb{F}_{19} : y^2 \equiv x^3 - 7x + 10 \pmod{19}\} \cup \{0\}.$$

Choose the point $p = (13, 8)$. Its multiples are as follows:

k	$k \cdot p$
0	0
1	$(13, 8)$
2	$(16, 17)$
3	$(18, 15)$
4	$(12, 1)$
5	$(5, 10)$
6	$(7, 0)$
7	$(5, 9)$
8	$(12, 18)$
9	$(18, 4)$
10	$(16, 2)$
11	$(13, 11)$
12	0

Thus p generates a cyclic subgroup of order 12.

As we can see from the example, each point generates some cyclic subgroup. The whole group of points on an elliptic curve does not have to be cyclic. On the other hand, the discrete logarithm construction requires a cyclic group. Therefore, for a given curve, we first compute its order. There is an efficient classical algorithm for this task, Schoof's algorithm [3]. Then we choose a prime divisor of the order and search for a point that generates a subgroup of this prime order.

The construction uses the following fact. For every point $g \in E$ we have

$$Ng = 0,$$

where $N = |E|$ is the order of the curve. Suppose that n is a prime divisor of N, so

$$N = hn.$$

Then

$$Ng = n(hg) = 0.$$

If $hg \neq 0$, then $g' = hg$ generates a cyclic subgroup whose order is n.

Remark. It is natural to choose a subgroup of prime order. If the order of a group is prime, then by Lagrange's theorem it has only two subgroups: the trivial subgroup and the group itself. Therefore any nonzero element of such a group is a generator.

Example. We will use the curve

$$E = E(\mathbb{F}_{97}) = \{(x, y) \in \mathbb{F}_{97} \times \mathbb{F}_{97} : y^2 \equiv x^3 - 7x + 10 \pmod{97}\} \cup \{0\}.$$

The order of this curve is

$$N = |E| = 82.$$

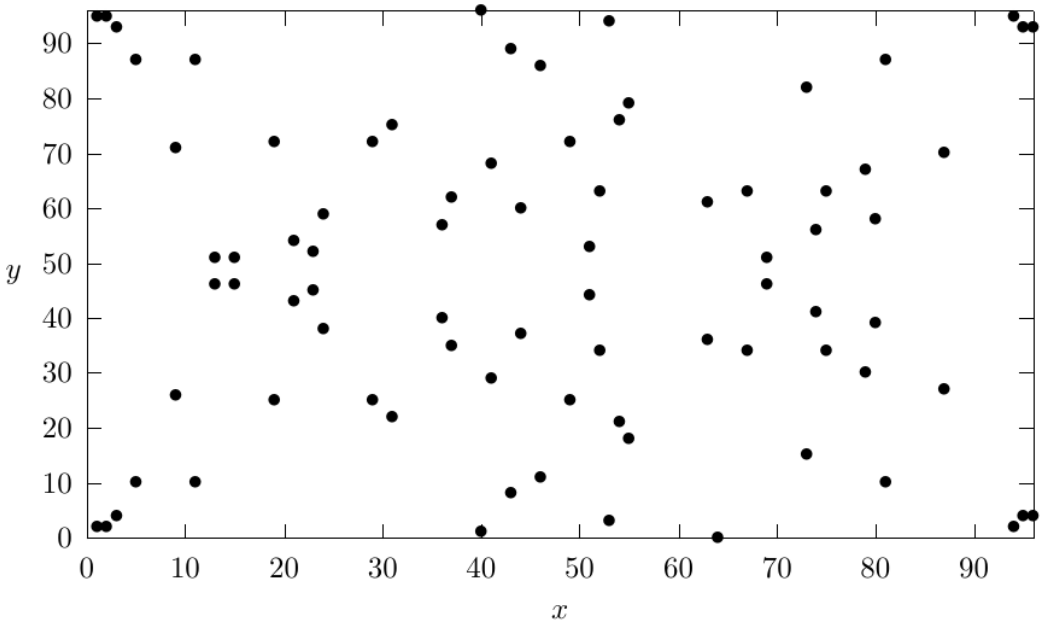

Figure 6: The curve $y^2 \equiv x^3 - 7x + 10 \pmod{97}$ over $\mathbb{F}_{97}$.

The number 82 has the prime divisor 41. Thus $h = 2$. Take $u = (1, 2)$. This point has order 82, so it is not a generator of the subgroup of order 41. We compute

$$g = hu = 2u = (96, 93).$$

The point g has order 41. This is the base point that we will use below.

Now we can state the elliptic-curve discrete logarithm problem. Given two points $a, b \in E(\mathbb{F}_p)$, we ask whether there is a number x such that

$$x \cdot a = b.$$

If such a number exists, then finding it is the discrete logarithm problem on elliptic curves.

4 The ECDH Algorithm

The ECDH algorithm is the Diffie-Hellman key exchange moved to elliptic curves. The public parameters are

$$(p, a, b, g, n, h),$$

where p, a, b define the curve

$$E(\mathbb{F}_p) = \{(x, y) : y^2 \equiv x^3 + ax + b \pmod{p}\} \cup \{0\},$$

the point g is the base point of order n, and h is the cofactor, i.e.

$$|E| = nh.$$

Alice chooses a private key

$$d_a \in \{1, \ldots, n-1\}$$

and computes the public key

$$A = d_a g.$$

Bob chooses a private key d_b and computes

$$B = d_b g.$$

After exchanging the public keys, both sides compute the same shared point:

$$K = d_a B = d_a d_b g = d_b A.$$

Example. Take the curve and the base point constructed above:

$$(p, a, b, g, n, h) = (97, -7, 10, (96, 93), 41, 2).$$

Alice chooses

$$d_a = 5.$$

Her public key is

$$A = d_a g = (37, 35).$$

Bob chooses

$$d_b = 15.$$

His public key is

$$B = d_b g = (15, 51).$$

Now Alice computes

$$d_a B = (46, 11),$$

and Bob computes

$$d_b A = (46, 11).$$

Thus both sides obtain the same shared point.

5 Shor's Algorithm

Now let us look at the same example from the point of view of Shor's discrete logarithm algorithm. We are given a base point g of order n and a point q in the cyclic subgroup generated by g. We have to find $x \in \mathbb{Z}/n\mathbb{Z}$ such that

$$xg = q. \tag{3}$$

In the ECDH example $q = A = (37, 35)$, and the hidden number is $x = d_a = 5$.

Consider the following auxiliary function:

$$f(x_1, x_2) = x_1 q + x_2 g. \tag{4}$$

Since $q = xg$, this function can be rewritten as

$$f(x_1, x_2) = (xx_1 + x_2)g.$$

This is the elliptic-curve analogue of the function used in Shor's algorithm for the ordinary discrete logarithm problem [2]. The ordinary group version, together with the two-dimensional Fourier derivation, is described in Quantum

Fourier Transform and Discrete Logarithms. Here the multiplicative group is replaced by the cyclic subgroup generated by the point g.

After the function is evaluated in a quantum register, the value register is measured. Suppose that the result is a point c from the subgroup generated by g. Then there is a number x_0 such that

$$c = x_0 g.$$

The remaining pairs (x_1, x_2) satisfy

$$xx_1 + x_2 \equiv x_0 \pmod{n}.$$

Thus, in the same way as in the ordinary discrete logarithm case, we can consider the function

$$f'(x_1, x_2) = \begin{cases} 1, & xx_1 + x_2 \equiv x_0 \pmod{n}, \\ 0, & xx_1 + x_2 \not\equiv x_0 \pmod{n}. \end{cases} \tag{5}$$

The Fourier image of this function contains information about the hidden number x. In the idealized case the maxima identify the period relation. In a finite sampled computation we obtain approximations, but the same idea remains.

Example. Use the ECDH public key

$$A = (37, 35)$$

and the base point

$$g = (96, 93).$$

We have

$$f(x_1, x_2) = x_1 A + x_2 g.$$

Let the measurement result be $c = g$. Then $x_0 = 1$, and the remaining pairs satisfy

$$xx_1 + x_2 \equiv 1 \pmod{41}.$$

For our small example the following pairs satisfy this relation:

x_1	x_2	$x_1 A + x_2 g$
0	1	g
7	7	g
8	2	g
15	8	g
16	3	g

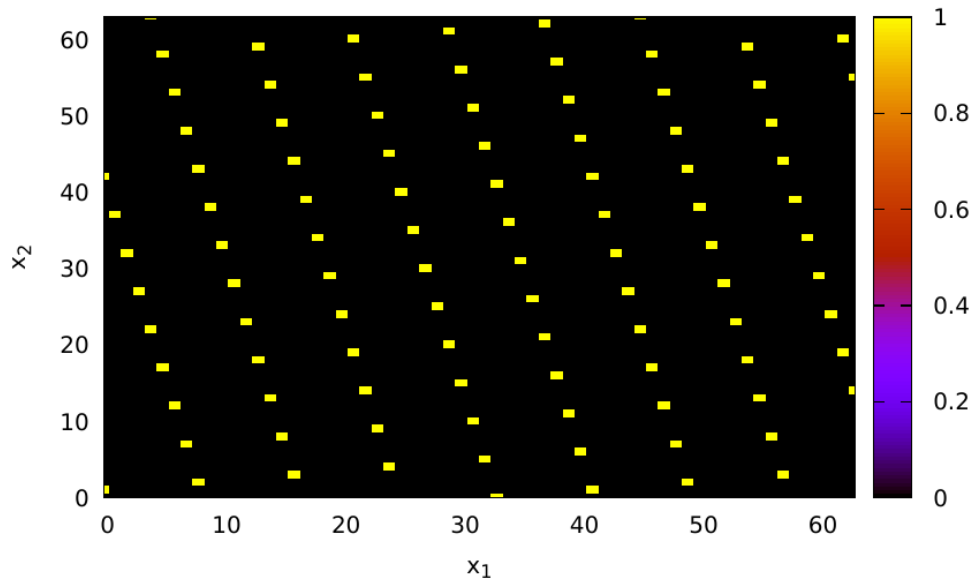

Figure 7: The points (x_1, x_2) that remain after measuring $c = g$, i.e. after selecting the level set $x_1 A + x_2 g = g$.

Take, for example, the pairs $(7, 7)$ and $(8, 2)$. Both correspond to the same measured point. Therefore

$$7x + 7 \equiv 8x + 2 \pmod{41}.$$

Subtracting the left-hand side from the right-hand side, we obtain

$$x(8 - 7) + (2 - 7) \equiv 0 \pmod{41}.$$

Hence

$$x \equiv 5 \pmod{41}.$$

This is exactly Alice's private key from the ECDH example.

We can also use the neighboring points $(8, 2)$ and $(16, 3)$. Their difference is $(8, 1)$, and therefore

$$8x + 1 \equiv 0 \pmod{41}.$$

Since $8^{-1} \equiv 36 \pmod{41}$, we get

$$x \equiv -36 \equiv 5 \pmod{41}.$$

In the original quantum algorithm, we do not inspect these pairs directly. Instead, the two-dimensional Fourier transform is applied to the register. For a sample size $M = 64$, the lower-left local maxima in this example are near

$$(8, 2), \quad (15, 3), \quad (24, 5).$$

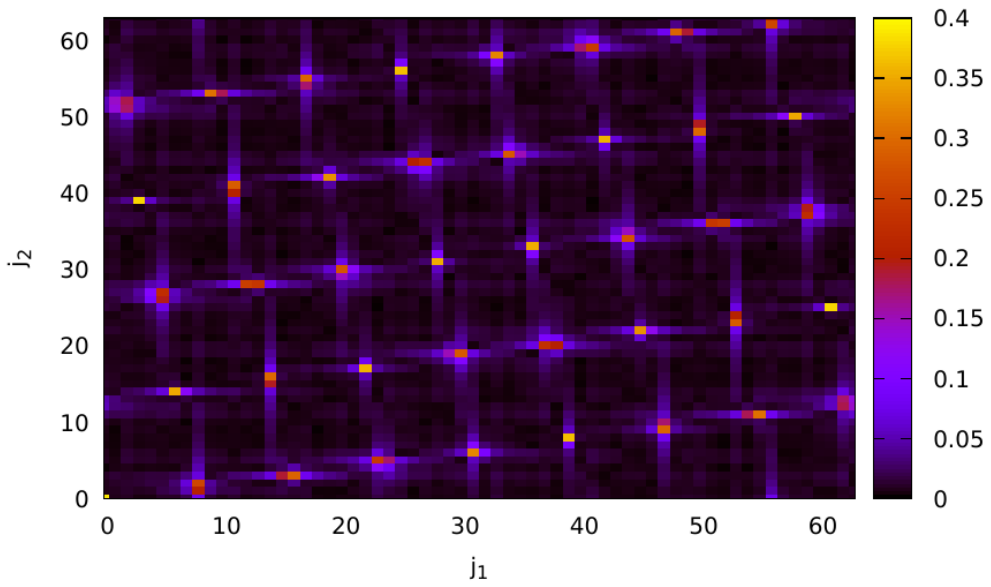

Figure 8: The Fourier image of the sampled function $f'(x_1, x_2)$ with $M = 64$. The lower-left local maxima give approximations to the hidden value $x = 5$.

The ratios

$$\frac{8}{2} = 4, \qquad \frac{15}{3} = 5, \qquad \frac{24}{5} = 4.8$$

give approximations to the hidden value $x = 5$. With large parameters the same mechanism is used to recover the discrete logarithm with high probability [1].

Conclusion

The ECDH protocol hides the private number d_a inside the equation

$$A = d_a g.$$

Classically, recovering d_a from A and g is the elliptic curve discrete logarithm problem. Shor's algorithm does not solve it by trying possible values of d_a.

Instead, it builds a function whose level sets encode the hidden linear relation

$$xx_1 + x_2 \equiv x_0 \pmod{n},$$

and then extracts this relation by Fourier analysis. For the small curve used in the article, this relation gives $x = 5$, the private key used by Alice.

References

[1] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[2] John Proos and Christof Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. *Quantum Information and Computation*, 3(4):317–344, 2003.

[3] René J. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of Computation*, 44:483–494, 1985.

[4] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE Computer Society, 1994.

[5] Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography*. Chapman & Hall/CRC, 2 edition, 2008.